



Лекция последняя

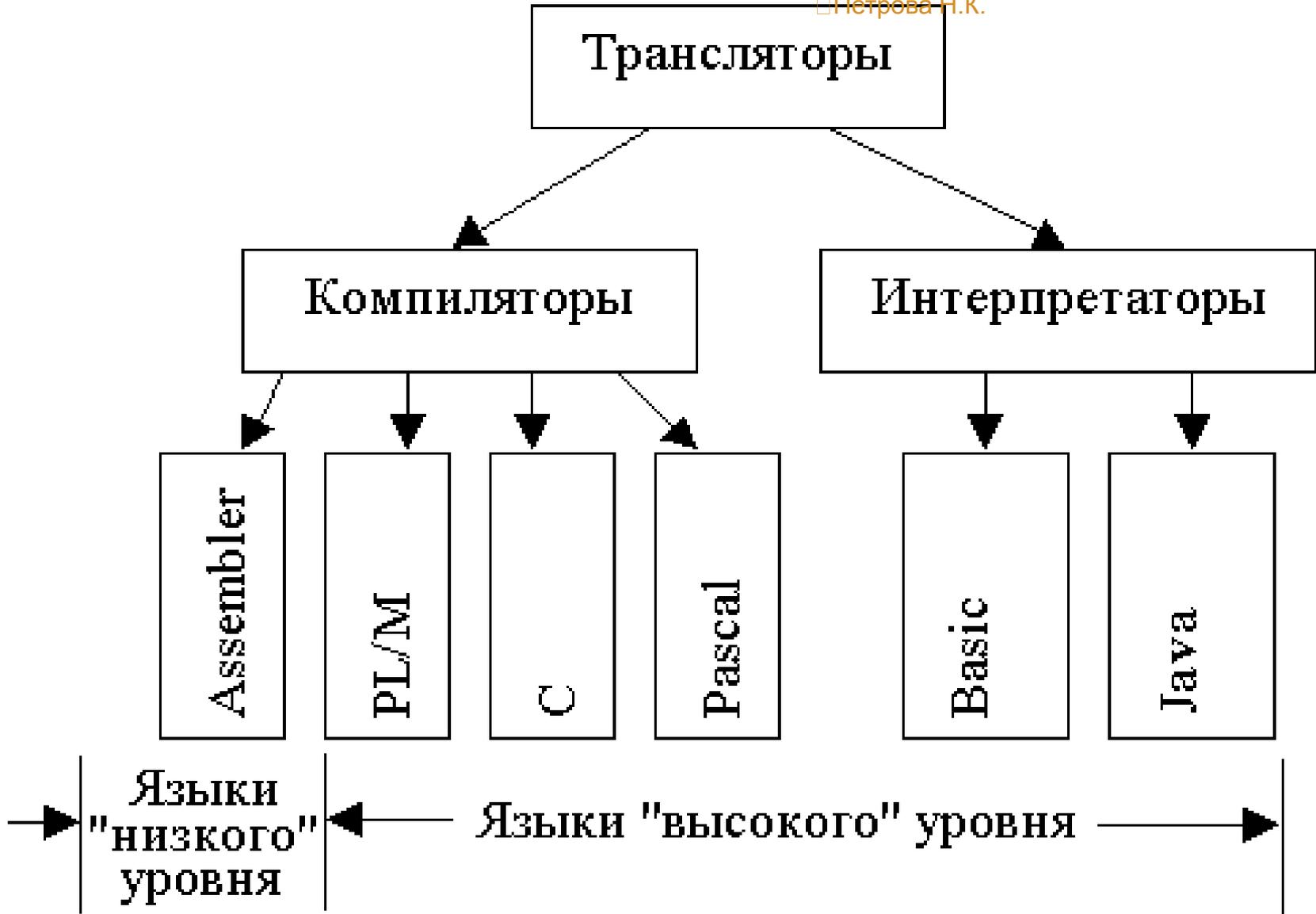
# ЯЗЫКИ ПРОГРАММИРОВАНИЯ

# ГРАДАЦИИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- ✘ Машинно-ориентированные, языки нижнего уровня: это языки, наборы операторов и изобразительные средства которых существенно зависят от особенностей ЭВМ (внутреннего языка, структуры памяти и т.д.). Программы, написанные на этих языках, *наиболее эффективно выполняются* на ЭВМ, т.к. учитываются все особенности конкретного процессора.
- ✘ Машинно–независимые или языки или языки высокого уровня: они ориентированы не на систему команд той или иной ЭВМ, а на систему операндов, характерных для записи определенного класса алгоритмов. Однако *программы, написанные на языках высокого уровня, занимают больше памяти и медленнее выполняются, чем программы на машинных языках.*

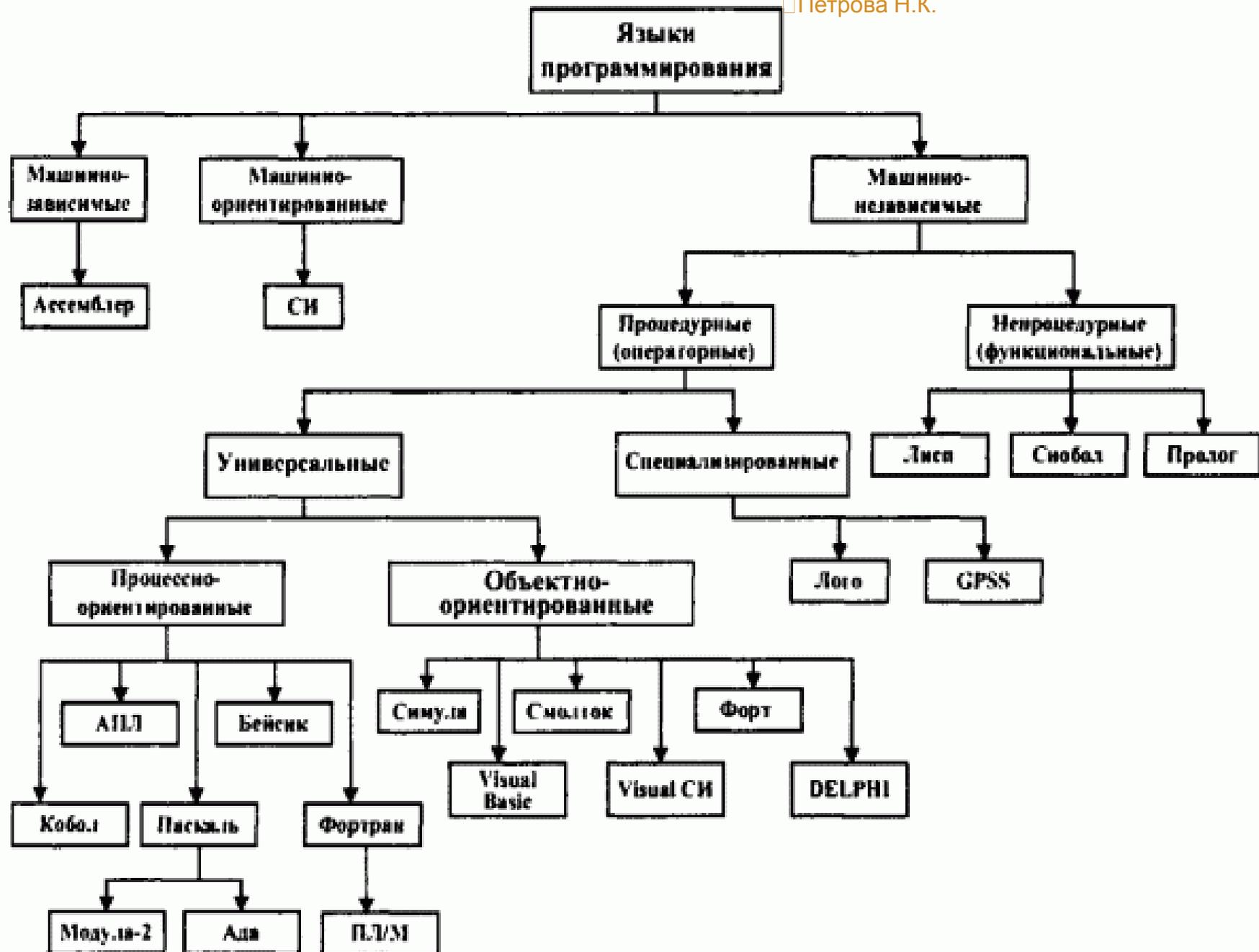
# ТРАНСЛЯТОРЫ: АССЕМБЛЕРЫ, ИНТЕРПРЕТАТОРЫ, КОМПИЛЯТОРЫ

- ✘ Транслятор – программа, переводящая текст программы с языка высокого уровня в язык машинных кодов.
- ✘ ассемблеры переводят программу, записанную на языке ассемблера, в программу на машинном языке;
- ✘ интерпретаторы - это транслятор, который производит *пооператорную обработку* и выполнение исходного кода программы;
- ✘ компиляторы – транслятор, который преобразует *всю программу в модуль на машинном языке*, после чего программа записывается в память компьютера и лишь потом исполняется)



# ПОКОЛЕНИЯ ЯЗЫКОВ

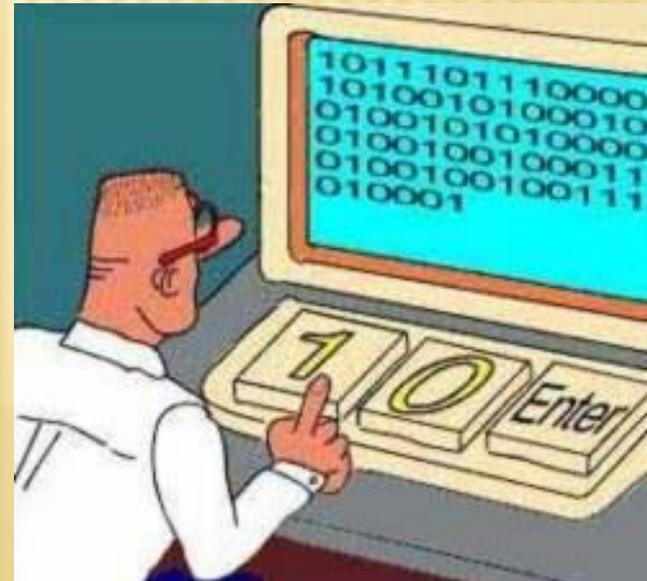
- ✘ – **языки первого поколения:** машинно–ориентированные с ручным управлением памяти на компьютерах первого поколения.
- ✘ – **языки второго поколения:** с мнемоническим представлением команд, так называемые автокоды.
- ✘ – **языки третьего поколения:** общего назначения, используемые для создания прикладных программ любого типа. Например, Бейсик, Кобол, Си и Паскаль.
- ✘ – **языки четвертого поколения:** усовершенствованные, разработанные для создания специальных прикладных программ, для управления базами данных. (SQL)
- ✘ – **языки программирования пятого поколения:** языки декларативные, объектно–ориентированные и визуальные. Например, Пролог, ЛИСП (используется для построения программ с использованием методов искусственного интеллекта), Си++, Visual Basic, Delphi.



# МАШИННЫЕ КОДЫ

- ✘ Для упрощения записи двоичная система записывалась в виде восьмеричной или шестнадцатеричной, но вводить коды нужно было все равно в системе исчисления с машинным основанием.
- ✘ Контроль ввода и конечного результата, а также поиск ошибок, осуществлялся по контрольным лампочкам на панели управления, либо по распечаткам.

Пример команд в машинных кодах для x86:  
V8 01 00 – поместить в регистр AX число 1.  
VV 02 00 – поместить в регистр VX число 2.  
01 D8 – сложить AX с VX, результат в AX.  
Эти же команды в двоичном виде:  
1011100000000000100000000  
1011101100000001000000000  
0000000111011000



## Пример команд на ассемблере:

mov ax,0001 – занести в регистр AX значение 1.  
mov bx,0002 – занести в регистр BX значение 2.  
add ax,bx – сложить AX и BX, результат – в AX.

# АССЕМБЛЕРЫ

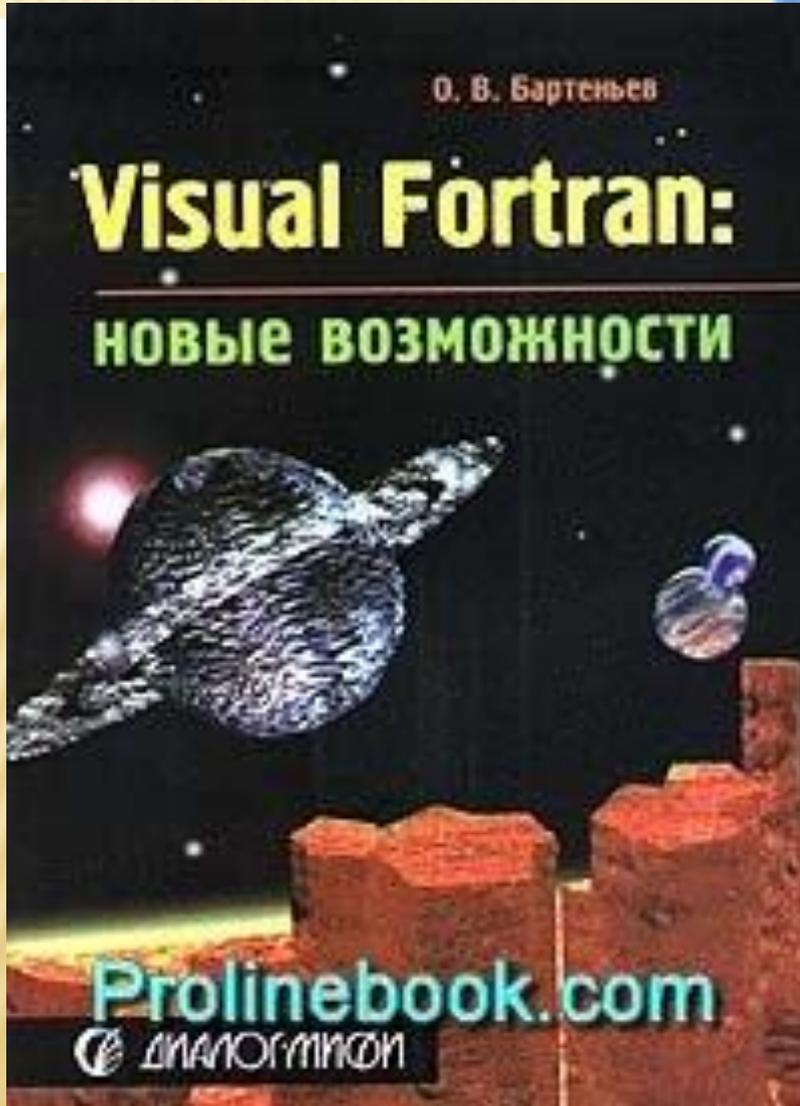
- ✘ *Ассемблер – это мнемоническая запись кодов центрального процессора.*
- ✘ Программу, написанную на ассемблере, переводит специальная программа-транслятор, заменяя каждую ассемблерную команду на соответствующий двоичный код.
- ✘ Набор полученных кодов полностью соответствует исходному набору ассемблерных команд, он *может быть непосредственно загружен в память машины и выполнен процессором.*
- ✘ Однако, поскольку каждый процессор использует только свой набор кодов, *такая программа не может быть непосредственно выполнена на машине с другим типом процессора.*

# ЯЗЫКИ ПРОГРАММИРОВАНИЯ ВЫСОКОГО УРОВНЯ

- ✘ •Fortran (1956), научные и инженерные расчеты.
- ✘ •COBOL (1959), для управления и бизнеса.
- ✘ •ALGOL (1958/1960), для записи алгоритмов.
- ✘ •PL/1 (1962), разработан для System/360.
- ✘ •BASIC (1964), для записи простых программ в учебных целях.



# FORTTRAN (1956) . ДЖОН БЕКУС



роки 150, 10 раз подряд.  
е переменной DO150I значения 1.10.

х и инженерных задачах.

рировали быстрый код,  
ссемблером, а сам язык  
пулярен среди ученых

IM ЯЗЫКОМ,  
множеством платформ, и  
олучившим широкое

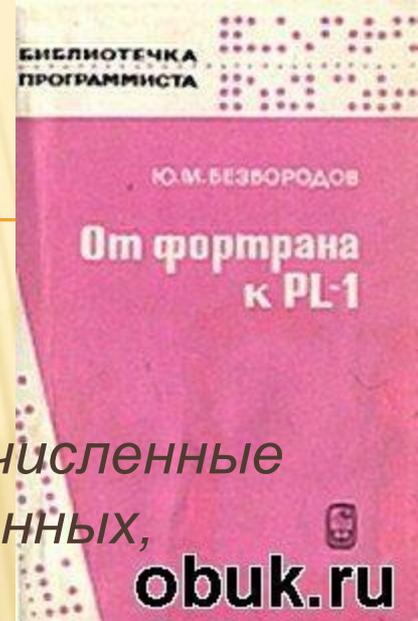
# COBOL (1960) ГРЕЙС ХОППЕР

- ✘ Реализована концепция аппаратно-независимого языка программирования
- ✘ Хоппер стремилась сделать язык программирования *похожим на обычный человеческий*.
- ✘ **Common Business-Oriented Language**, универсальный язык, ориентированный на задачи бизнеса.
- ✘ *Читабельность, самодокументируемость и простота языка, наряду с поддержкой разработчиками и аппаратной независимостью*, сделали его очень популярным для применения в коммерческих компаниях и правительственных организациях в 60–70-х годах.
- ✘ *Особенно эффективно в Коболе производятся простые арифметические операции с большими массивами данных*, что довольно часто приходится делать в бухгалтерских расчетах.
- ✘ Существование языка стимулировало развитие многих других языков программирования высокого уровня, которые используют квази-английский синтаксис



# ALGOL (1958)

- ✘ ALGOL (**ALGO**rithmic **L**anguage, алгоритмический язык программирования) - **универсальный язык** для широкого круга применений.
- ✘ Планировалось, что он избежит ряда проблем языка Fortran, но выявленные новые проблемы привели к тому, что «классический» ALGOL вышел только в 1960 г.
- ✘ Алгол почти на 30 лет стал стандартом описания алгоритмов в учебниках и научных работах, а многие языки, разработанные впоследствии, содержат многие идеи и решения, взятые из Алгола:
  - *концепция операторных алгоритмических языков с заранее фиксированными типами данных и блочной структурой,*
  - появилась возможность разработки *отдельных модулей* проекта независимо друг от друга,
  - реализации *вызовов фрагментов кода программы,*
  - способов *передачи параметров между процедурами и функциями,*
  - появилась *рекурсия.*
  - появилось представление о программе как о блочной структуре, а не

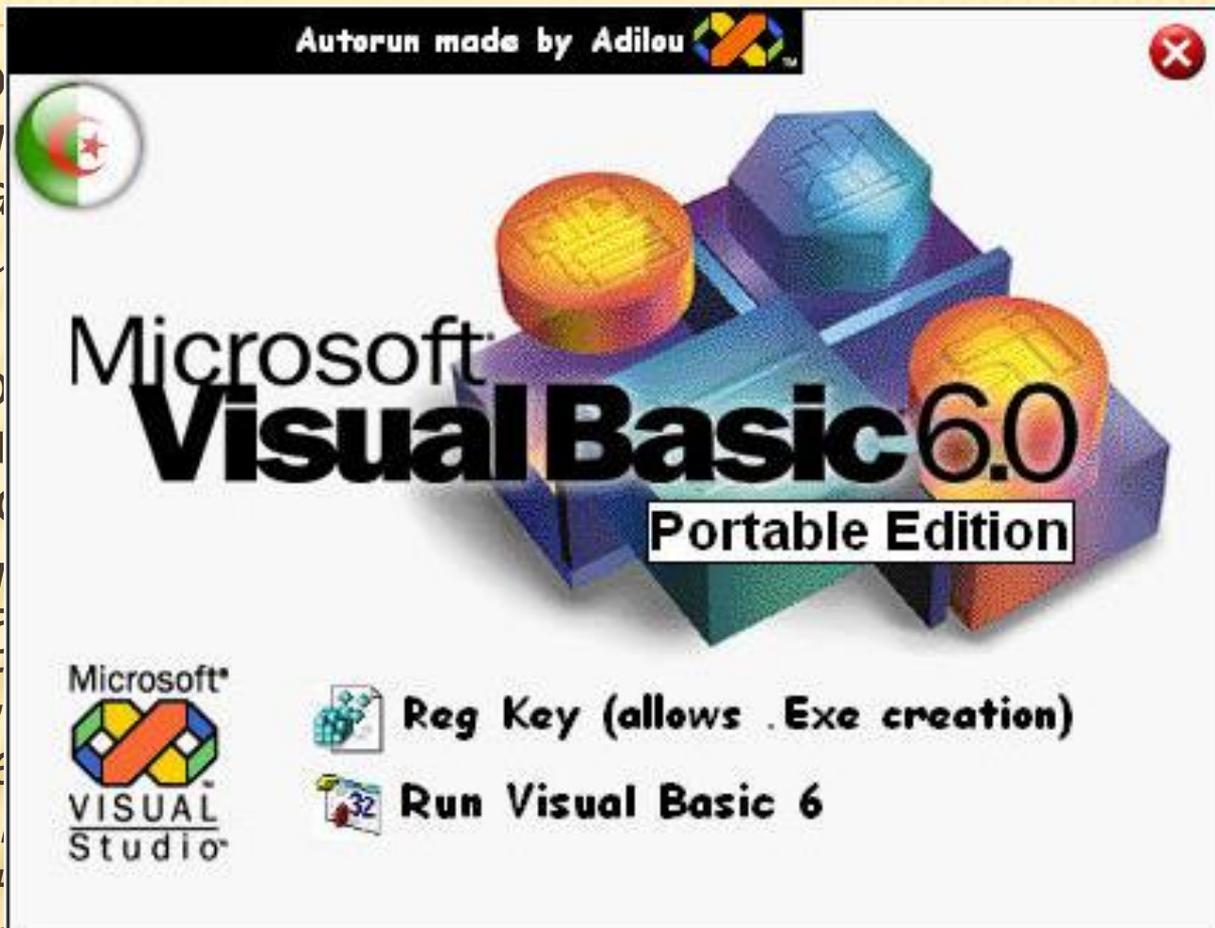


# PL/1 (1966) IBM

- ✗ **PL/1 (Programming Language One)**
- ✗ Язык обеспечивал обработку данных и *научные численные вычисления, рекурсии, связанные структуры данных, символьные и битовые строки.*
- ✗ Синтаксис языка был похож на английский язык и подходил для описания сложных форматов данных, с широким набором функций, доступных для проверки и манипулировать ими.
- ✗ Вследствие попытки **удовлетворить нужды всех пользователей** язык получился перегруженным возможностями и концепциями. Это привело к сложностям при разработке компиляторов и тому, что генерируемый код оказывался далеким от оптимального, проигрывая в математических задачах Фортрану.

# BASIC (1964) ТОМАС КУРЦ И ДЖОН ДЖОРДЖ КЕМЕНИ

Лекция 8. Языки программирования  
Петрова К.К.



ы в  
новом  
ий, даже  
tion Code,  
ЦИХ),  
ЛЬНОЙ  
ьшинство  
SAVE,  
ились с  
е  
к данных или

- × Цель р систем програ гумани
- × BASIC универ позвол подгот
- × По син команд PRINT,
- × Програ перфо выполн операторов.
- × Разработанный язык оказался очень универсальным, в сравнении с другими – небольшим, а сам компилятор разработчики сделали бесплатным, что сделало BASIC очень популярным.

# Первый микрокомпьютер

---

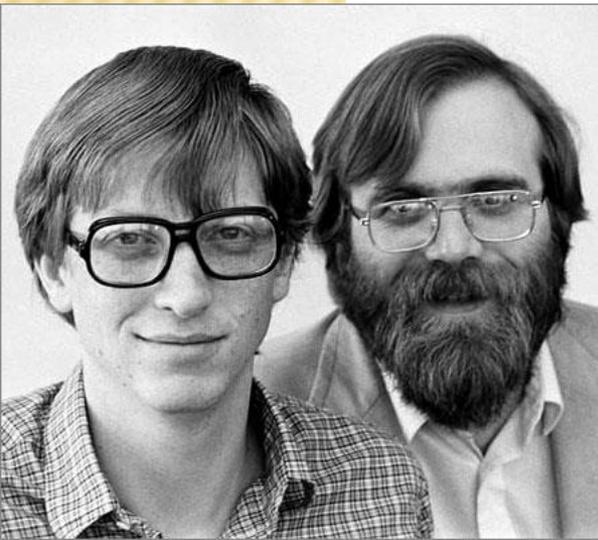
## 1974. Альтаир-8800 (Э. Робертс)

- комплект для сборки
- процессор *Intel 8080*
- частота 2 МГц
- память 256 байт



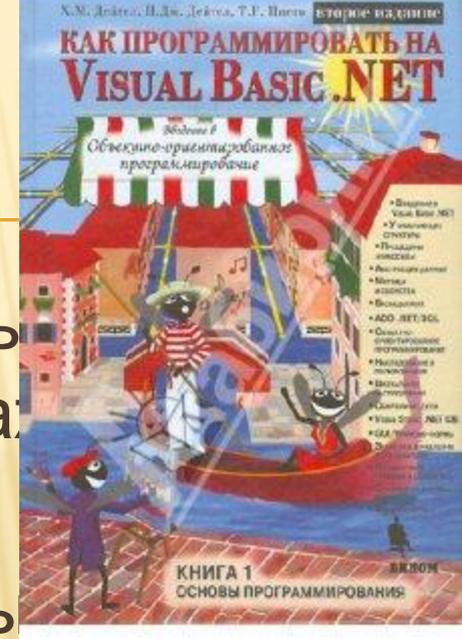
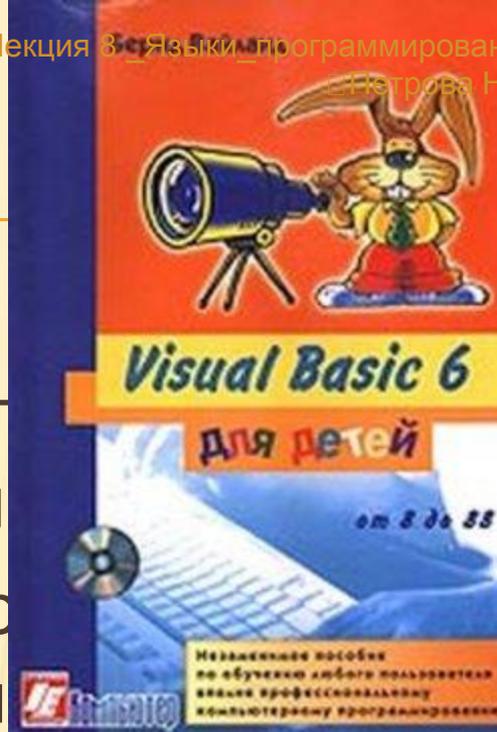
## 1975. Б. Гейтс и П. Аллен

транслятор языка  
Альтаир-Бейсик



# BASIC

- ✗ Вопреки ожиданиям «языка для начинающих» оказался популярным среди разработчиков бизнес-приложений.
- ✗ Разработано много рунимов, в том числе и для мобильных устройств. В настоящее время он используется в виде интерпретатора VBScript или в виде языка Visual Basic .NET.
- ✗ Из недостатков языка отмечают широко применяющийся оператор GOTO, а так же то, что программы через интерпретатор часто работали быстрее, чем откомпилированные.



# УНИВЕРСАЛЬНЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

- ✘ •Pascal (1970), улучшение Алгола и задачи обучения.
- ✘ •C (1973), для системного программирования.
- ✘ •Ada (1980), для встроенных систем реального времени.
- ✘ •Lisp (1958), Prolog (1972), обработка символьной информации.

# PASCAL (1970) НИКОЛАС ВИРТ

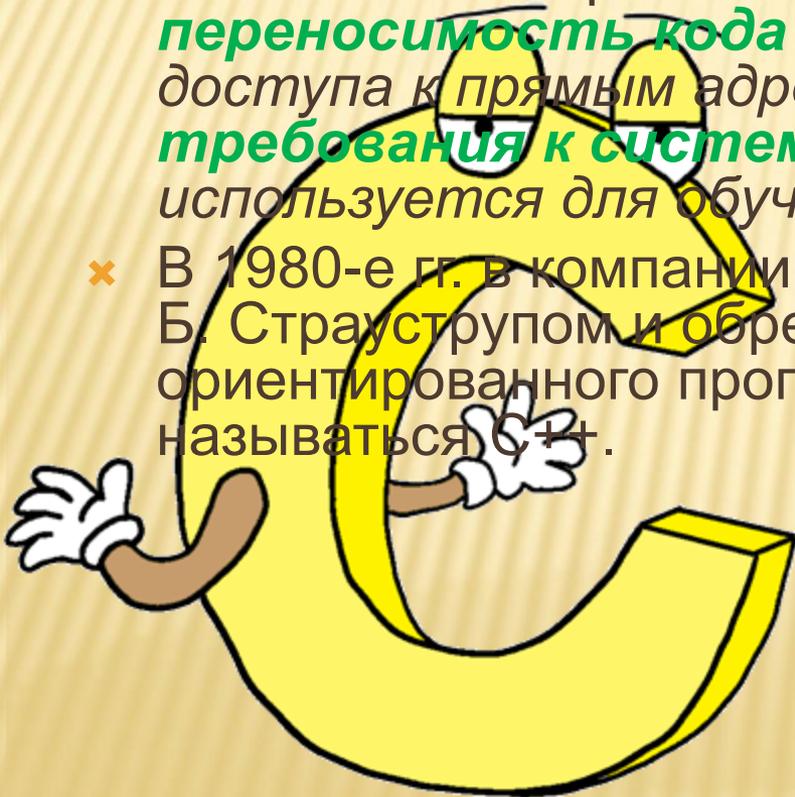


- ✘ Язык Паскаль требовал от программиста определения всех переменных в отдельной секции в начале программы.
- ✘ *Так как эти определения задавались явным образом, то в программах появлялось сравнительно немного ошибок, и их было проще понять и исправить разработчику. Это сделало Паскаль популярным при создании больших программ.*
- ✘ *Паскаль был основным языком высокого уровня в первых компьютерах Apple, многие программы были написаны на объектно-ориентированных версиях языка – Delphi или Object Pascal.*

# C (1973) ДЕННИС РИТЧИ



- ✘ Язык был призван обеспечить компиляцию с помощью простого компилятора, низкоуровневый доступ к памяти, максимальную близость к машинным командам и минимизировать время на отладку
- ✘ Оказались востребованы такие его свойства, как **переносимость кода** и эффективность, возможность доступа к прямым адресам оборудования, а также **низкие требования к системным ресурсам**. Так же язык часто используется для обучения программированию.
- ✘ В 1980-е гг. в компании Bell Labs язык был усовершенствован Б. Страуструпом и обрел поддержку объектно-ориентированного программирования. Такой язык стал называться C++.





# ADA (1980)

- ✘ Ада (в честь Ады Лавлейс), можно характеризовать как Паскаль, расширенный добавлением новых элементов.
- ✘ В результате получился существенно более сложный язык. Причем **было запрещено выпускать к нему компиляторы, не прошедшие официальное тестирование**, поэтому его применение далеко за пределы военных встроенных систем не вышло.
- ✘ Особенности Ада являются *строгая типизация, модульность, проверка на наличие ошибок на этапе компиляции и в реальном режиме, поддержка параллельной обработки данных и обработка исключений*.
- ✘ По этим причинам Ада широко используется в системах, где сбой может привести к серьезным последствиям – в авионике (управление Boeing 777), в системах вооружения (в т.ч. ядерное оружие), космических кораблях и пр

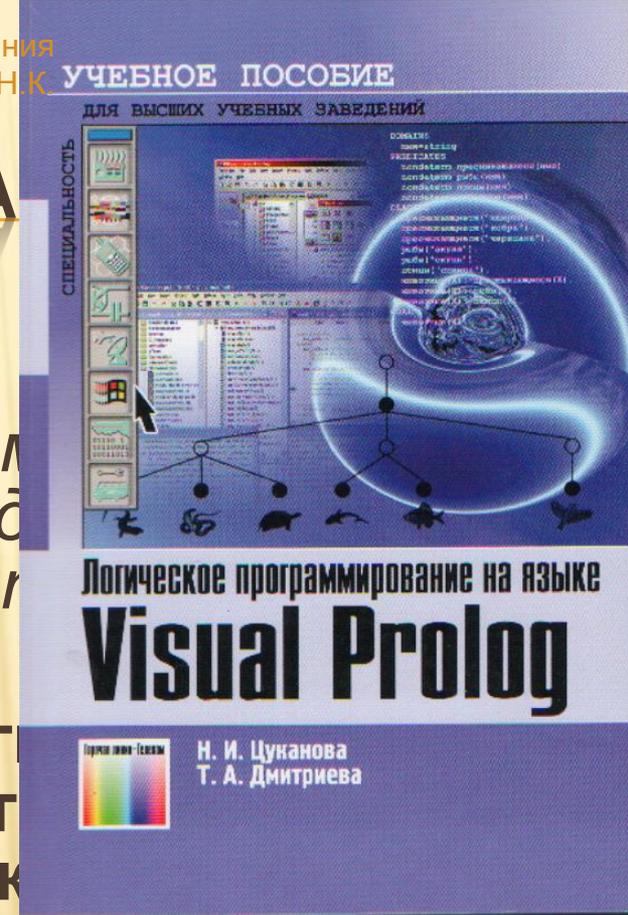


## **LISP (1958) ДЖ. МАККАРТИ (США)**

- ✘ Языки обработки символьной информации, Lisp (List Processing Language)
- ✘ В нем и программы, и данные представляются в *виде связанного списка символов*, в итоге программа может управлять исходным кодом, как списком данных.
- ✘ Он стал основой ряда программных реализаций интеллектуальных систем и дал толчок к разработке множества специализированных языков **искусственного интеллекта и языков представлений знаний**.

# PROLOG (1972) АЛАНом КОЛМА

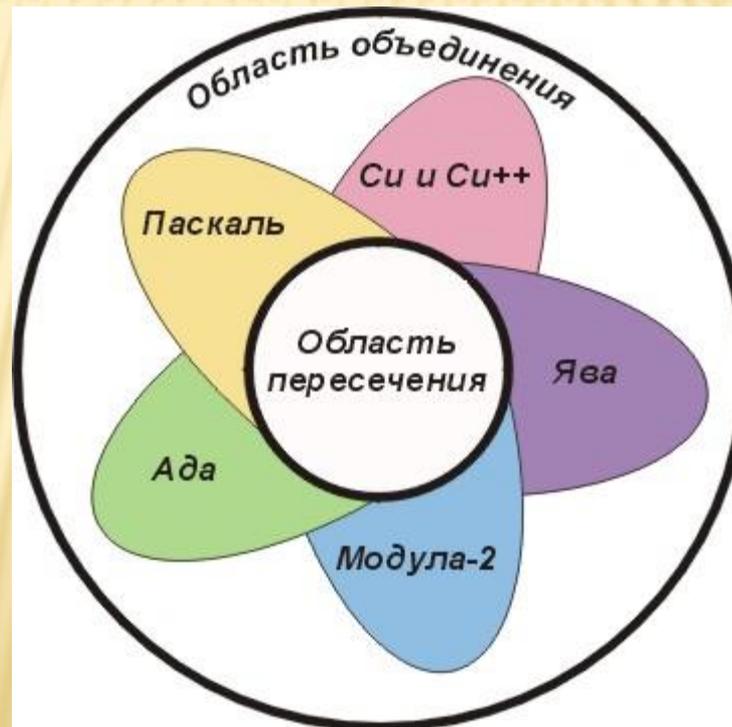
- ✗ Prolog (Programming in Logic).
- ✗ Это *единственный язык программирования* основанный на логическом выводе поставленной задачи, что роднит его с искусственным интеллектом.
- ✗ В отличие от большинства других языков перед ним достаточно поставить задачу, а затем он сам будет искать решение.
- ✗ Из-за необычности своей структуры он использовался при решении достаточно нестандартных задач, поэтому оказался распространен не так широко, как другие языки.



# LANGUAGE PEOPLE



**СЕМАНТИКУ КАЖДОГО ЯЗЫКА  
ПРОГРАММИРОВАНИЯ МОЖНО УСЛОВНО  
ПОДЕЛИТЬ НА "ОБЛАСТЬ ПЕРЕСЕЧЕНИЯ"  
(ОБЩИЕ ДЛЯ ВСЕХ ЯЗЫКОВ КОНСТРУКЦИИ) И  
"ОБЛАСТЬ ОБЪЕДИНЕНИЯ" (СПЕЦИФИЧЕСКИЕ  
ДЛЯ ДАННОГО ЯЗЫКА КОНСТРУКЦИИ).**



# ТЕНДЕНЦИИ В РАЗВИТИИ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

- ✘ К началу 1980-х годов были разработаны **принципы объектно-ориентированного программирования**, что создало возможности для появления новых языков, таких как C++ или Object Pascal.
- ✘ Еще одну такую возможность создал в середине 90-х гг. быстрый рост Интернета. Среди последних, например, **язык Perl**, изначально разработанный в 1987 г. в Unix для создания *сценариев*. Он получил широкое распространение в динамических веб-сайтах.
- ✘ Язык **Java**, созданный для применения во *встраиваемых системах*, стали использовать для веб-программирования.
- ✘ Во многом такие языки не являются принципиально новыми, а представляются уточнением и развитием существующих.
- ✘ *В целом, современные языки и парадигмы программирования в значительной степени основаны на семействе языков программирования C.*

# ИСТОРИЯ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Лекция 8\_Языки\_программирования

□Петрова Н.К.

Алгоритм, записанный на «понятном» компьютеру языке программирования, называется **программой**.

Языки программирования на платформе .NET

Языки программирования для компьютерных сетей

Языки объектно-ориентированного программирования

Алгоритмические языки программирования

Языки программирования высокого уровня

JavaScript, 1995

Visual J#, 2003

Java, 1995

Visual C#, 2003

C, 1971

C++, 1983

Object Pascal, 1989

Delphi, 1995

Pascal, 1970

Turbo Delphi, 2006

BASIC, 1964

QBASIC, 1975

Visual Basic, 1991

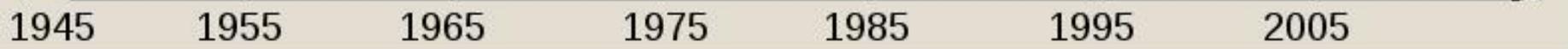
Visual Basic .Net, 2003

COBOL, 1959

FORTRAN, 1954

Ассемблер, 1951

Машинный язык, 1945



**ЛЕКЦИЯ ЗАКОНЧЕНА**